Programmieren in Java

Vorlesung 03: Schleifen

Prof. Dr. Peter Thiemann

Albert-Ludwigs-Universität Freiburg, Germany

SS 2017

Inhalt

Codequalität

Standards auf der Webseite

Einlesen von Konsole und Dateien

Lösungsmuster mit java.util.Scanner

```
1  package linenumbers;
2  import java.util.Scanner;
3  public class Main {
4    public static void main(String[] args) {
5      // get a scanner
6      while (sc.hasNextLine()) {
7        String line = sc.nextLine();
8        // do s.t. with the line
9      }
10     sc.close();
11   }
12 }
```

Ende der Eingabe

- Dateiende
- magic key (CMD-D) in IntelliJ

Verwenden von Argumenten

```
1 package linenumbers;
2 public class Main {
3   public static void main(String[] args) {
4     int nrOfParameters = args.length;
5     String arg1 = args[0];
6     String arg2 = args[1];
7     doSomething(arg1, arg2);
8   }
9 }
```

Standardmuster: Schleife mit Array

```
1 package printarguments;
2 public class Main {
3    public static void main(String[] args) {
4       int nrOfParameters = args.length;
5       for (int i = 0; i < nrOfParameters; i++) {
6          System.out.println((i+1) + "tes Argument: " + args[i]);
7       }
8    }
9 }
```

Alternative: Schleife mit Array

```
1 package printarguments;
2 public class Main {
3   public static void main(String[] args) {
4     for (String s : args) {
5       System.out.println(s);
6     }
7   }
8 }
```

- Index ist nicht verfügbar
- Vorteil: einfacher, Nachteil: wenn Index notwendig ist ...

Zweidimensionale Felder

```
1    int[][] matrix = new int[ROWS][COLS];
2    for (int i = 0; i < ROWS; i++) {
3      for (int j = 0; j < COLS; j++) {
4        matrix[i][j] = initialize(i, j);
5      }
6    }
```

Zur Übung

- Implementiere Matrixmultiplikation

Verarbeiten der Zeichen eines Strings

```
1  public static void traverse(String s) {
2    for (int i = 0; i < s.length(); i++) {
3      char c = s.charAt(i);
4      // do s.t. with c
5    }
6  }
```

Umwandlung zwischen `char` und `int`

Geschieht am einfachsten durch einen **type cast**:

```
1  char c;
2  int codeOfC = (int) c;
3  // do s.t. with codeOfC
4  char newC = (char) codeOfC;
```

Nichtfunktionale Anforderungen

P. Thiemann, L. Fennell Sommersemester 2017

Programmieren in Java

`http://proglang.informatik.uni-freiburg.de/teaching/java/2017/`

second-highest

Die zweithöchste Zahl

Woche 03 Aufgabe 2/4

Herausgabe: 2017-05-08
Abgabe: 2017-05-19

Achtung: beachten Sie unbedingt die allgemeinen Hinweise zur Abgabe auf der Homepage.

Project `second-highest`
Package `secondhighest`
Klassen

`Main`
`public static int secondHighest(int[] numbers)`

Implementieren Sie die Funktion `secondHighest`, die in einem `int`-Array die *zweit*größte Zahl findet und zurückgibt. Kann die Zahl nicht gefunden werden, soll `Integer.MIN_VALUE` zurückgegeben werden.

Ihre Implementierung sollte das Ergebnis in nur einem Durchlauf des Arrays `numbers` berechnen (also ohne das Array beispielsweise zu sortieren). Diese Anforderung zählt zur Code-Qualität!

Beispielaufruf 01: `Main.secondHighest(new int[]{1, 5, 5, 6, 1, 7})` ergibt `6` (als `int`)

Verwendung von StringBuilder

```
1  StringBuilder sb = new StringBuilder(42); // erwartete Zeichenzahl
2  //
3  char c:
4  sb.append(c);
5  sb.append(Character.toChars(c));
6  //
7  String s = sb.toString();
```

- StringBuilder: veränderlicher String
- Erzeugen mit **new** StringBuilder()
- Anhängen von Zeichen, Strings, Zahlen, etc mit append()
- Umwandlen in String mit toString()

Fragen